GENERATIVE EMBEDDINGS BASED ON RICIAN MIXTURES
Application to Kernel-based Discriminative Classification of Magnetic Resonance Images

Anna C. Carli[1], Mário A. T. Figueiredo[2], Manuele Bicego[1,3] and Vittorio Murino[1,3]
[1]*Dipartimento di Informatica, Università di Verona, Verona, Italy*
[2]*Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal*
[3]*Istituto Italiano di Tecnologia (IIT), Genova, Italy*

Keywords: Discriminative learning, Magnetic resonance images, Generative embedding, Information theory, Kernels, Rice distributions, Finite mixtures, EM algorithm.

Abstract: Most approaches to classifier learning for structured objects (such as images or sequences) are based on probabilistic generative models. On the other hand, state-of-the-art classifiers for vectorial data are learned discriminatively. In recent years, these two dual paradigms have been combined via the use of generative embeddings (of which the Fisher kernel is arguably the best known example); these embeddings are mappings from the object space into a fixed dimensional score space, induced by a generative model learned from data, on which a (maybe kernel-based) discriminative approach can then be used.
This paper proposes a new semi-parametric approach to build generative embeddings for classification of magnetic resonance images (MRI). Based on the fact that MRI data is well described by Rice distributions, we propose to use Rician mixtures as the underlying generative model, based on which several different generative embeddings are built. These embeddings yield vectorial representations on which kernel-based support vector machines (SVM) can be trained for classification. Concerning the choice of kernel, we adopt the recently proposed nonextensive information theoretic kernels.
The methodology proposed was tested on a challenging classification task, which consists in classifying MRI images as belonging to schizophrenic or non-schizophrenic human subjects. The classification is based on a set of regions of interest (ROIs) in each image, with the classifiers corresponding to each ROI being combined via boosting. The experimental results show that the proposed methodology outperforms the previous state-of-the-art methods on the same dataset.

1 INTRODUCTION

Most approaches to learning classifiers belong to one of two paradigms: generative and discriminative (Ng and Jordan, 2002; Rubinstein and Hastie, 1997). Generative approaches are based on probabilistic class models and *a priori* class probabilities, learnt from training data and combined via Bayes law to yield posterior probability estimates. Discriminative learning methods aim at learning class boundaries or posterior class probabilities directly from data, without relying on generative class models.

In the past decade, several hybrid generative-discriminative approaches have been proposed with the goal of taking advantage of the best of both paradigms (Jaakkola and Haussler, 1999; Lasserre et al., 2006). In this context, the so-called generative score space methods (or generative embeddings) have stimulated significant interest. The key idea is to exploit a generative model to map the objects to be classified into a feature space, where discriminative techniques, namely kernel-based ones, can be used. This is particularly suitable to deal with non-vectorial data (strings, trees, images), since it maps objects (maybe of different dimensions) into a fixed dimension space.

The seminal work on generative embeddings is arguably the Fisher kernel (Jaakkola and Haussler, 1999). In that work, the features of a given object are the derivatives of the log-likelihood under the assumed generative model, with respect to the model parameters, computed at that object. Other examples

We acknowledge financial support from the FET programme within EU FP7, under the SIMBAD project (contract 213250).

C. Carli A., A. T. Figueiredo M., Bicego M. and Murino V. (2012).
GENERATIVE EMBEDDINGS BASED ON RICIAN MIXTURES - Application to Kernel-based Discriminative Classification of Magnetic Resonance Images.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 113-122
DOI: 10.5220/0003790801130122

of generative embeddings have been more recently proposed (Bosch et al., 2006; Perina et al., 2009).

In this paper, we exploit generative embeddings to tackle a challenging classification task: based on a set of regions of interest (ROIs) of a magnetic resonance image (MRI), classify the patient as suffering, or not, from schizophrenia (Cheng et al., 2009a). We build on the knowledge of the fact that MRI data is well modeled by Rician distributions (Gudbjartsson and Patz, 1994), and propose several generative embeddings based on Rician mixture models. Concerning the kernels used in the obtained feature space, we adopt the nonextensive information theoretic kernels recently proposed by (Martins et al., 2009). An SVM classifier is learnt for each ROI. Finally, an optimal combination of these SVM classifiers is learnt via the AdaBoost algorithm (Freund and Schapire, 1997). The experimental results reported show that the proposed methodology outperforms the previous state-of-the-art on the same dataset.

The paper is organized as follows. Section 2 addresses the problem of estimating Rician finite mixtures using the expectation-maximization (EM) algorithm. In Section 3, we propose several generative embeddings based on the Rician mixture model. Section 4 briefly reviews the information theoretic kernels proposed by (Martins et al., 2009), while Section 5 described SVM combination via boosting. Finally, Section 6 reports the experimental results on the magnetic resonance (MR) image categorization problem.

2 RICIAN MIXTURE FITTING VIA THE EM ALGORITHM

2.1 The EM Algorithm

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is the most common approach for computing the maximum likelihood estimate (MLE) of the parameters of a finite mixture. In this section, we briefly review how EM is used to estimate a mixture of Rician distributions. A Rician probability density function (Rice, 1944) has the form

$$f_R(y;\nu,\sigma) = \frac{y}{\sigma^2} e^{-\frac{y^2+\nu^2}{2\sigma^2}} I_0\left(\frac{y\nu}{\sigma^2}\right), \qquad (1)$$

for $y > 0$, and zero for $y \leq 0$, where ν is the magnitude parameter, σ is the noise parameter, and $I_0(z)$ denotes the 0-th order modified Bessel function of the first kind (Abramowitz and Stegun, 1972)

$$I_0(z) = \frac{1}{2\pi}\int_0^{2\pi} e^{z\cos\phi} d\phi. \qquad (2)$$

A finite mixture of Rician distributions, with g components, is thus

$$f(y;\Psi) = \sum_{i=1}^{g} \pi_i f_R\left(y;\nu_i,\sigma_i^2\right), \qquad (3)$$

where the π_i's, $i = 1,\ldots,g$, are nonnegative quantities that sum to one (the so-called mixing proportions or weights), $\theta_i = (\nu_i,\sigma_i^2)$ is the pair of parameters of component i, and $\Psi = (\pi_1,\ldots,\pi_{g-1},\theta_1,\ldots,\theta_g)$ is the vector of all the parameters of the mixture model.

Let $Y = \{y_1,\ldots,y_n\}$ be a random sample of size n, assumed to have been generated independently by a mixture of the form (3) and consider the goal of obtaining an MLE of Ψ, that is, $\widehat{\Psi} = \arg\max_\Psi L(\Psi)$, where

$$L(\Psi,Y) = \sum_{j=1}^{n} \log f(y_j;\Psi) = \sum_{j=1}^{n} \log \sum_{i=1}^{g} \pi_i f_R\left(y_j;\nu_i,\sigma_i^2\right). \qquad (4)$$

As is common in EM, let $\mathbf{z}_j \in \{0,1\}^g$ be a g-dimensional hidden/missing binary label vector associated to observation y_j, such that $z_{ji} = 1$ if and only if y_j was generated by the i-th mixture component. The so-called complete data is $\{(y_1,\mathbf{z}_1),\ldots,(y_n,\mathbf{z}_n)$ and the corresponding complete loglikelihood for Ψ, $\log L_c(\Psi)$, is given by

$$L_c(\Psi,Y,Z) = \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ji}\left\{\log\pi_i + \log f_R(y_j;\theta_i)\right\} \qquad (5)$$

where $Z = \{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$.

The EM algorithm proceeds iteratively in two steps. The E-step computes the conditional expectation (with respect to the missing labels Z), of the complete loglikelihood given the observed data y and the current parameter estimate $\widehat{\Psi}^{(k)}$,

$$Q(\Psi;\Psi^{(k)}) := \mathbb{E}_Z\left[L_c(\Psi,Y,Z)|Y,\widehat{\Psi}^{(k)})\right]. \qquad (6)$$

Since the complete-data log likelihood is linear in the unobservable data z_{ij} (as is clear in (5)), this reduces to computing the conditional expectation of hidden variables and plugging these into the complete loglikelihood. These conditional expectations are well known and equal to the posterior probability that the j-th sample was generated by the ith component of the mixture; denoting this quantity as w_{ji}, we have

$$w_{ji} = \frac{\pi_i f(y_j;\theta_i^{(k)})}{\sum_{h=1}^{g}\pi_h^{(k)} f(y_j;\theta_h^{(k)})}, \qquad (7)$$

for $i = 1,\ldots,g$ and $j = 1,\ldots,n$. It follows that the conditional expectation of the complete loglikelihood (6) becomes

$$Q(\Psi;\Psi^{(k)}) = \sum_{i=1}^{g}\sum_{j=1}^{n} w_{ji}\left\{\log\pi_i + \log f(y_j;\theta_i)\right\}. \qquad (8)$$

The M-step obtains an updated parameter estimate $\Psi^{(k+1)}$ by maximizing $Q(\Psi;\Psi^{(k)})$ with respect to Ψ over the parameter space Ω. The updated estimates of the mixing proportions $\pi_i^{(k+1)}$ are well-known to be given by

$$\pi_i^{(k+1)} = \frac{1}{n}\sum_{j=1}^{n} w_{ji}. \tag{9}$$

2.2 Updating the Parameters of the Rician Components

Updating the estimate of $\theta_i = (\nu_i, \sigma_i^2)$ requires solving

$$\sum_{i=1}^{g}\sum_{j=1}^{n} w_{ji}\nabla_\theta \log f_R(y_j;\theta) = 0, \tag{10}$$

where ∇_θ denotes the gradient with respect to θ. In the following proposition (proved in the appendix), we provide an explicit solution of (10) for the Rician mixture.

Proposition 2.1. *The updated estimate* $\widehat{\theta}_i^{(k+1)} = (\widehat{\nu}_i^{(k+1)}, (\widehat{\sigma}_i^2)^{(k+1)})$*, that is, the solution of* (10)*, is*

$$\widehat{\nu}_i^{(k+1)} = \frac{1}{\sum_{j=1}^{n} w_{ji}}\sum_{j=1}^{n} w_{ji} y_j\, \phi\Big(\frac{y_j \nu_i^{(k)}}{\sigma_i^{2(k)}}\Big) \tag{11}$$

and

$$(\widehat{\sigma}_i^2)^{(k+1)}) = \frac{0.5}{\sum_{j=1}^{n} w_{ji}}\sum_{j=1}^{n} w_{ji}\left(y_j^2 + \nu_i^{(k+1)^2} - 2y_j\nu_i^{(k+1)}\phi\Big(\frac{y_j\nu_i^{(k)}}{\sigma_i^{2(k)}}\Big)\right) \tag{12}$$

where

$$\phi(u) = \frac{I_1(u)}{I_0(u)}. \tag{13}$$

3 GENERATIVE EMBEDDINGS BASED ON RICIAN MIXTURES

This section introduces several generative embeddings for images based on the Rician mixture model. Let $X_s = \{y_1^s,\ldots,y_{N_s}^s\}$, for $s = 1,\ldots,S$, be a set of images, each belonging to one of R classes. Each image X_s is modeled simply as a bag of N_s strictly positive pixels $y_j^s \in \mathbb{R}_{++}$, for $j = 1,\ldots,N_s$. Each image is mapped into a finite-dimensional Hilbert space (the so–called *feature space*) using the Rician mixture generative model, as explained next.

Based on a K-components Rician mixture with parameters Ψ, the posterior probability that y_j^s (the j-th pixel of the s-th image) belongs to the i-th component of the mixture is

$$w_i(y_j^s;\Psi) = \frac{\pi_i f(y_j^s;\theta_i)}{\sum_{k=1}^{K}\pi_k f(y_j^s;\theta_k)}, \tag{14}$$

as used in the E-step (7). Based on (14), different generative embeddings can be defined, as shown in Definitions 3.1, 3.2, and 3.3.

Definition 3.1. *If a single Rician mixture* Ψ *is estimated for the S images, the embedding of an image* $X = \{y_1,\ldots,y_N\}$ *is a K-dimensional vector given by*

$$\widetilde{\mathbf{e}}^{\,single}(X;\Psi) = \frac{1}{N}\left[\sum_{j=1}^{N} w_1(y_j;\Psi),\ldots,\sum_{j=1}^{N} w_K(y_j;\Psi)\right]^T. \tag{15}$$

Definition 3.2. *If a set of R Rician mixtures (one per class) is estimated,* $\{\Psi_1,\ldots,\Psi_R\}$*, each with K components, the embedding of an image* $X = \{y_1,\ldots,y_N\}$ *is a* (KR)*-dimensional vector given by*

$$\widetilde{\mathbf{e}}(X;\Psi_1,\ldots,\Psi_R) = \frac{1}{N_s}\left[\left(\widetilde{\mathbf{e}}^{\,single}(X;\Psi_1)\right)^T,\ldots,\left(\widetilde{\mathbf{e}}^{\,single}(X;\Psi_R)\right)^T\right]^T. \tag{16}$$

Other possible embeddings and their generalizations are introduced in the following definition.

Definition 3.3. *We will also consider the two following K-dimensional embeddings, defined for an arbitrary image* $X = \{y_1,\ldots,y_N\}$ *as*

$$\overline{\mathbf{e}}^{\,single}(X;\Psi) = \frac{1}{N}\sum_{j=1}^{N}\Big[\pi_1 f(y_j;\theta_1),\ldots,\pi_K f(y_j;\theta_K)\Big]^T$$

and

$$\widehat{\mathbf{e}}^{\,single}(X;\Psi) = \frac{1}{N}\sum_{j=1}^{N}\Big[f(y_j;\theta_1),\ldots,f(y_j;\theta_1)\Big]^T,$$

as well as their (KR)*-dimensional generalizations to the case in which a Rician mixture is estimated for each of the R classes,*

$$\overline{\mathbf{e}}(X;\Psi_1,\ldots,\Psi_R) = \left[\left(\overline{\mathbf{e}}^{\,single}(X;\Psi_1)\right)^T,\ldots,\left(\overline{\mathbf{e}}^{\,single}(X;\Psi_R)\right)^T\right]^T$$

and

$$\widehat{\mathbf{e}}(X;\Psi_1,\ldots,\Psi_R) = \left[\left(\widehat{\mathbf{e}}^{\,single}(X;\Psi_1)\right)^T,\ldots,\left(\widehat{\mathbf{e}}^{\,single}(X;\Psi_R)\right)^T\right]^T.$$

4 NONEXTENSIVE INFORMATION THEORETIC KERNELS ON MEASURES

This section briefly reviews the information theoretic kernels proposed in (Martins et al., 2009), introducing notation which will be useful later on.

4.1 Suyari's Entropies

Begin by recalling that both the Shannon-Boltzmann-Gibbs (SBG) and the Tsallis entropies are particular cases of functions $S_{q,\phi}$ following Suyari's axioms (Suyari, 2004). Let Δ^{n-1} be the standard probability simplex and $q \geq 0$ be a fixed scalar (the *entropic index*). The function $S_{q,\phi} : \Delta^{n-1} \to \mathbb{R}$ has the form

$$S_{q,\phi}(p_1,\cdots,p_n) = \begin{cases} \frac{k}{\phi(q)}\left(1-\sum_{i=1}^n p_i^q\right) & \text{if } q \neq 1 \\ -k\sum_{i=1}^n p_i \ln p_i & \text{if } q = 1 \end{cases} \tag{17}$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a continuous function with properties stated in (Suyari, 2004), and $k > 0$ an arbitrary constant, henceforth set to $k = 1$. For $q = 1$, we recover the SBG entropy,

$$S_{1,\phi}(p_1,\cdots,p_n) = H(p_1,\cdots,p_n) = -\sum_{i=1}^n p_i \ln p_i,$$

while setting $\phi(q) = q-1$ yields the Tsallis entropy

$$S_q(p_1,\cdots,p_n) = \frac{1}{q-1}\left(1-\sum_{i=1}^n p_i^q\right) = -\sum_{i=1}^n p_i^q ln_q p_i,$$

where $\ln_q(x) = \frac{(x^{1-q}-1)}{1-q}$ is the q-logarithmic function.

4.2 Jensen-Shannon (JS) Divergence

Consider two measure spaces $(\mathcal{X},\mathcal{M},\nu)$, and $(\mathcal{T},\mathcal{J},\tau)$, where the second is used to index the first. Let H denote the SBG entropy, and consider the random variables $T \in \mathcal{T}$ and $X \in \mathcal{X}$, with densities $\pi(t)$ and $p(x) \triangleq \int_{\mathcal{T}} p(x|t)\pi(t)$. The Jensen divergence (Martins et al., 2009) is defined as

$$J^\pi(p) \triangleq J_H^\pi(p) = H(\mathbb{E}[p]) - \mathbb{E}[H(p)]. \tag{18}$$

When $\mathcal{X}$ and $\mathcal{T}$ are finite with $|\mathcal{T}| = m$, $J_H^\pi(p_1,\cdots,p_m)$ is called the *Jensen-Shannon (JS) divergence* of $p_1,\cdots,p_m$, with weights $\pi_1,\cdots,\pi_m$ (Burbea and Rao, 1982), (Lin, 1991). In particular, if $|\mathcal{T}| = 2$ and $\pi = (1/2,1/2)$, p may be seen as a random distribution whose value on $\{p_1,p_2\}$ is chosen tossing a fair coin. In this case, $J^{(1/2,1/2)} = JS(p_1,p_2)$, where

$$JS(p_1,p_2) \triangleq H\left(\frac{p_1+p_2}{2}\right) - \frac{H(p_1)+H(p_2)}{2},$$

which will be used in Section 4.4 to define JS kernels.

4.3 Jensen-Tsallis (JT) q–Differences

Notice that Tsallis' entropy can be written as

$$S_q(X) = -\mathbb{E}_q[\ln_q p(X)],$$

where $\mathbb{E}_q$ denotes the *unnormalized q–expectation*, which, for a discrete random variable $X \in \mathcal{X}$ with probability mass function $p : \mathcal{X} \to \mathbb{R}$, is defined as

$$\mathbb{E}_q[X] \triangleq \sum_{x\in\mathcal{X}} x\,p(x)^q;$$

(of course, $\mathbb{E}_1[X]$ is the standard expectation).

As in Section 4.2, consider two random variables $T \in \mathcal{T}$ and $X \in \mathcal{X}$, with densities $\pi(t)$ and $p(x) \triangleq \int_{\mathcal{T}} p(x|t)\pi(t)$. The Jensen q-difference (nonextensive analogue of (18)) (Martins et al., 2009) is

$$T_q^\pi(p) = S_q(\mathbb{E}[p]) - \mathbb{E}_q[S_q(p)].$$

If $\mathcal{X}$ and $\mathcal{T}$ are finite with $|\mathcal{T}| = m$, $T_q^\pi(p_1,\cdots,p_m)$ is called the *Jensen-Tsallis (JT) q-difference* of $p_1,\cdots,p_m$, with weights $\pi_1,\cdots,\pi_m$. In particular, if $|\mathcal{T}| = 2$ and $\pi = (1/2,1/2)$, define $T_q = T_q^{1/2,1/2}$

$$T_q(p_1,p_2) = S_q\left(\frac{p_1+p_2}{2}\right) - \frac{S_q(p_1)+s_q(p_2)}{2},$$

which will be used in Section 4.4 to define JT kernels. Naturally, T_1 coincides with the JS divergence.

4.4 Jensen-Shannon and Tsallis Kernels

The JS and JT differences underlie the kernels proposed in (Martins et al., 2009), which can be defined for normalized or unnormalized measures.

Definition 4.1 (Weighted Jensen-Tsallis kernels)**.** *Let μ_1 and μ_2 be two (not necessarily probability) measures; the kernel $\widetilde{k}_q$ is defined as*

$$\widetilde{k}_q(\mu_1,\mu_2) \triangleq \left(S_q(\pi) - T_q^\pi(p_1,p_2)\right)(\omega_1+\omega_2)^q$$

where $p_1 = \frac{\mu_1}{\omega_1}$ and $p_2 = \frac{\mu_2}{\omega_2}$ are the normalized counterparts of μ_1 and μ_2, with corresponding total masses ω_1 and ω_2, and $\pi = (\omega_1+\omega_2)^{-1}[\omega_1,\omega_2]$. The kernel k_q is defined as

$$k_q(\mu_1,\mu_2) \triangleq S_q(\pi) - T_q^\pi(p_1,p_2)$$

Notice that if $\omega_1 = \omega_2$, $\widetilde{k}_q$ and k_q coincide up to a scale factor. For $q = 1$, k_q is the so-called Jensen-Shannon kernel, $k_{JS}(p_1,p_2) = \ln 2 - JS(p_1,p_2)$.

The following proposition characterizes these kernels in terms of positive definiteness, a crucial aspect for their use in support vector machines (SVM).

Proposition 4.1. *The kernel $\widetilde{k}_q$ is positive definite (pd), for $q \in [0,2]$. The kernel k_q is pd, for $q \in [0,1]$. The kernel k_{JS} is pd.*

5 COMBINING SVM CLASSIFIERS VIA BOOSTING

The final building block of our approach to MR image classification is a way to combine the classifiers working on each of the several regions of interest (ROI). For that end, we adopt the Adaboost algorithm (Freund and Schapire, 1997), which we now briefly review. In the description of AdaBoost in Algorithm 5.1, each (weak) classifiers $G_m(x)$, $m = 1,\ldots,M$, each corresponding to one of the M regions.

Algorithm 5.1: AdaBoost (Freund and Schapire, 1997).

1. Initialize weights $p_i = 1/S$, $i = 1,\ldots,S$.
2. For $m = 1$ to M:
   (a) Learn classifier $G_m(x)$ with current weights.
   (b) Compute weighted error rate:
   $$\text{err}_m = \frac{\sum_{i=1}^{S} p_i 1_{(y_i \neq G_m(x_i))}}{\sum_{i=1}^{S} p_i}.$$
   (c) Compute $\gamma_m = \log(1-\text{err}_m) - \log(\text{err}_m)$.
   (d) $p_i \leftarrow p_i \cdot \exp(\gamma_m 1_{(y_i \neq G_m(x_i))})$, $i = 1,\ldots,S$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \gamma_m G_m(x)\right]$.

Each boosting step requires learning a classifier by minimizing a weighted criterion, that is, with weights $p_1,\ldots,p_S$ corresponding to each training observations (y_i, X_i), $i = 1,\ldots,S$. In our case, the classifier G_m is a weighted version of the SVM classifier corresponding to the m-th ROI, *i.e.*, the SVM classifier whose kernel function is built on the Rician mixture estimated for that ROI. To take into account these weights, the optimization problem solved by the SVM learning algorithm requires a modification: the penalty on the slack variable ξ_i corresponding to the example X_i is set to be proportional to the weight p_i. The corresponding modified 1-norm SVM optimization problem (Cristianini and Shawe-Taylor, 2000), (Schölkopf and Smola, 2002) is

$$\begin{aligned} \min_{\xi,\beta,\beta_0} \quad & \langle \beta,\beta\rangle + C\textstyle\sum_{i=1}^{S} p_i\xi_i \qquad (19)\\ \text{s.t.} \quad & y_i\left(\langle \beta,\phi(X_i)\rangle + \beta_0\right) \geq 1-\xi_i, \quad i = 1,\ldots,S \\ & \xi_i \geq 0, \quad i = 1,\ldots,S. \end{aligned}$$

The Lagrangian for problem (19) is

$$\begin{aligned} L_p(\beta,\beta_0,\xi,\alpha,\mu) &= \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{S} p_i\xi_i \\ &- \sum_{i=1}^{S} \alpha_i\left[y_i(\langle\phi(X_i),\beta\rangle + \beta_0) - (1-\xi_i)\right] - \sum_{i=1}^{S}\mu_i\xi_i \end{aligned} \qquad (20)$$

with $\alpha_i \geq 0$ and $\mu_i \geq 0$. By minimizing L_p with respect to β, β_0, ξ_i and μ_i, $i = 1,\ldots,S$, the Lagrange dual problem results

$$\begin{aligned} \max_{\alpha} \quad & \textstyle\sum_{i=1}^{S}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{S}\alpha_i\alpha_j y_i y_j k(X_i,X_j) \qquad (21)\\ \text{s.t.} \quad & 0 \leq \alpha_i \leq p_i C \\ & \textstyle\sum_{i=1}^{S}\alpha_i y_i = 0\,. \end{aligned}$$

Notice that each α_i is constrained to be less or equal to $p_i C$ rather than C while the objective function in (21) is the same as the original 1-norm dual problem (Cristianini and Shawe-Taylor, 2000), (Schölkopf and Smola, 2002). As a consequence, if p_i is close to zero, so is α_i, thus contributing very weakly to the definition of the optimal hyperplane, which is still given by

$$f(X,\alpha^*,\beta_0) = \sum_{i=1}^{S} y_i\,\alpha_i^*\,k(X_i,X) + \beta_0^*. \qquad (22)$$

6 EXPERIMENTS

Let us begin this section with a summary of the proposed approach. The training data consists of set of images, each containing a set of M regions of interest (ROI) and labeled as belonging to a schizophrenic or non-schizophrenic patient. For each ROI of the set of training images, either a single Rician mixture or two Rician mixtures (one for each class) are estimated and used to embed the data on a Hilbert space, as described in Section 3. On the Hilbert space for each ROI, one of the information theoretic kernels described in Section 4 is used. Finally, a set of M (one per ROI) SVM classifiers is obtained by the AdaBoost algorithm described in Section 5; the final classifier is the one resulting at the last step of Algorithm 5.1.

The baselines against which we compare the proposed approach are SVM classifiers with linear kernels (LK) and Gaussian radial basis function kernels (GRBFK) built on the same generative embeddings. SVM training is carried out using the LIBSVM package (http://www.csie.ntu.edu.tw/∼cjlin/libsvm). The underlying Rician mixtures were estimated using the EM algorithm described in Section 2, with K (the number of components) selected using the criterion proposed in (Figueiredo and Jain, 2002); this leads to numbers in the $[4,6]$ range. We tested the generative embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$ proposed in Section 3, both in the single-mixture and R-mixtures versions.

The dataset contains 124 images (64 patients and 60 controls), each with the following 14 ROIs (7 pairs): Amygdala (1-Left, 2-Right), Dorso-lateral PreFrontal Cortex (3-Left, 4-Right), Entorhinal Cortex (5-Left, 6-Right), Heschl's Gyrus (7-Left, 8-Right), Hippocampus (9-Left, 10-Right), Superior

Table 1: Mean accuracy for the best values of q and C for the SVM classifiers learnt on ROI 2, 4, 6 respectively, using one Rician mixture per class with $K = 4,5,6$ components and embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$.

ROI	2			4			6		
No. of components	4	5	6	4	5	6	4	5	6
Embedding $\widetilde{\mathbf{e}}$									
Linear	54.84	53.06	53.39	60.16	60	60	57.26	58.23	58.23
RBF	59.52	60.16	**62.26**	60.81	60.81	61.13	65.32	65.16	64.48
Jensen-Shannon	58.87	58.39	59.84	60.81	58.55	60.32	67.42	66.61	65.48
Jensen-Tsallis	59.35	60	60.97	**62.42**	59.84	**62.42**	67.58	67.42	65.97
Weighted JT $\widetilde{k}_q$	59.35	59.84	60.97	61.13	60.32	61.94	**67.74**	67.26	66.29
Weighted JT k_q	59.35	59.19	59.84	**62.42**	59.84	**62.42**	67.58	66.94	65.97
Embedding $\overline{\mathbf{e}}$									
Linear	53.06	51.94	51.94	58.87	58.23	57.74	56.45	58.55	57.74
RBF	61.94	62.26	**63.39**	59.84	60.48	**60.97**	64.03	63.39	63.55
Jensen-Shannon	60	61.45	60.32	57.74	57.74	57.26	64.84	65.48	65.81
Jensen-Tsallis	61.45	61.45	62.9	**60.48**	60.16	60	67.1	67.58	66.61
Weighted JT $\widetilde{k}_q$	62.58	62.26	62.1	57.9	58.06	58.87	66.13	65.97	65
Weighted JT k_q	61.77	61.45	**63.23**	56.94	58.06	57.09	66.45	66.94	**67.74**
Embedding $\widehat{\mathbf{e}}$									
Linear	52.74	53.55	55.65	58.39	58.06	58.55	57.1	57.26	57.1
RBF	61.94	62.1	**63.39**	60.32	**60.65**	60.32	65.81	64.84	65.16
Jensen-Shannon	60.48	60.32	60.97	57.74	57.74	57.9	65	66.45	65.97
Jensen-Tsallis	60.97	61.13	**63.39**	59.52	**60.16**	59.52	66.76	**68.06**	66.29
Weighted JT $\widetilde{k}_q$	62.1	62.58	62.42	58.39	57.9	58.55	64.08	65	65.65
Weighted JT k_q	61.45	61.45	62.42	57.74	57.74	59.84	65.32	66.13	67.74

Temporal Gyrus (11-Left, 12-Right), Thalamus (13-Left, 14-Right). To evaluate the classifiers, the dataset was split 50%-50% into training and test subsets and 10 runs were performed.

SVM classifiers were trained for each individual ROI (without the boosting-based combination), and the conclusion was that ROI 10 leads to the best accuracy (see Tables 1, 2, 3). The accuracy is robust to the number of components of the mixture. The best performances over q and C are reported. For the GRBFK, the best performance over the width parameter and over C are reported. Mean accuracies are plotted in Figure 1 as a function of q for the best value of C and as a function of C for the best value of q, for the generative embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$, with 2 (one per class) Rician mixtures each with 4 components. The results with a single mixture are very similar, thus omitted. For $q > 1$, the results shown for the weighted JT kernel (which is positive definite only for $q \in [0,1]$) correspond to $q = 1$. These results show that the proposed generative embeddings lead to comparable performances. The information theoretic kernels outperform the LK and GRBFK. Namely, the best performances are obtained with the JT and weighted JT kernels, for all ROIs. The standard error of the mean is less than 0.006.

Results obtained by combining the SVM classifiers with the AdaBoost algorithm are shown in Table 4 for the generative embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$. These results show that the proposed approach outperforms state-of-the-art methods for ROIs intensity histograms for this dataset, see (Cheng et al., 2009a), (Cheng et al., 2009b), (Ulas et al., 2010), (Ulas et al., 2011).

7 CONCLUSIONS

In this paper, we have proposed a new approach for building generative embeddings for kernel-based classification of magnetic resonance images (MRI) by exploiting the Rician distribution that characterizes MR images. Using generative embeddings, the images to be classified are mapped onto a Hilbert space, where kernel-based techniques can be used. Concerning the choice of kernel, we have adopt the recently proposed nonextensive information theoretic kernels. The proposed approach was tested on a challenging classification task: classifying subjects as suffering, or not, from schizophrenia on the basis of a set of regions of interest (ROIs) in each image. To this purpose, an SVM classifier for each ROI is learnt. Finally, we propose to combine the SVM classifiers via

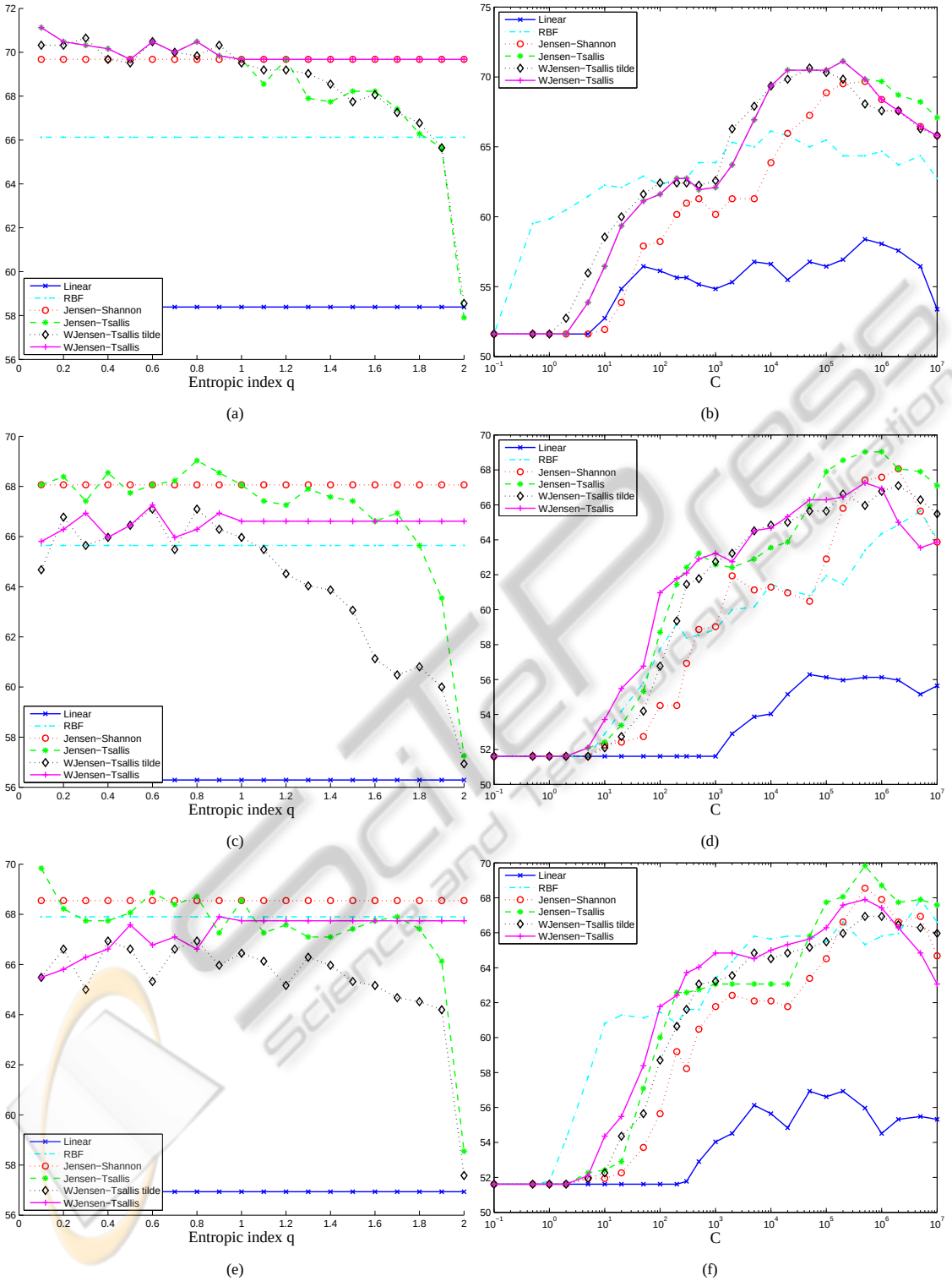

Figure 1: Mean accuracy on 10 runs as a function of q (best C) and as a function of C (best q) for the SVM classifier learnt on ROI 10 using one Rician mixture per class with $K = 4$ components and embeddings $\widetilde{\mathbf{e}}$ ((a), (b)), $\overline{\mathbf{e}}$ ((c), (d)) and $\widehat{\mathbf{e}}$ ((e), (f)).

Table 2: Mean accuracy for the best values of q and C for the SVM classifiers learnt on ROI 8, 12, 14 respectively, using one Rician mixture per class with $K = 4, 5, 6$ components and embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$.

ROI	8			12			14		
No. of components	4	5	6	4	5	6	4	5	6
Embedding $\widetilde{\mathbf{e}}$									
Linear	62.58	60.32	59.52	58.39	60.65	59.35	55.32	55	55.48
RBF	65.48	65.32	64.03	65.97	65.32	63.71	61.94	**62.74**	61.13
Jensen-Shannon	65.32	65	64.84	64.35	64.84	64.52	62.42	61.45	60.16
Jensen-Tsallis	66.45	66.13	65.65	66.13	**66.94**	64.68	62.58	62.1	61.45
Weighted JT $\widetilde{k}_q$	**67.26**	66.77	65.65	66.13	66.29	65	**62.74**	61.94	61.45
Weighted JT k_q	66.45	65.65	65.65	66.13	**66.94**	64.68	62.58	62.1	61.45
Embedding $\overline{\mathbf{e}}$									
Linear	59.35	60.16	59.19	58.23	59.03	57.26	55	54.84	54.84
RBF	63.71	64.68	63.23	62.42	62.9	62.9	62.1	63.55	63.06
Jensen-Shannon	63.71	64.68	63.23	60.65	61.94	62.1	66.61	65.98	65.32
Jensen-Tsallis	64.68	64.84	64.68	62.58	**64.84**	63.71	**67.9**	66.61	66.29
Weighted JT $\widetilde{k}_q$	**65.16**	64.19	63.23	63.87	64.19	62.9	65.48	64.84	63.87
Weighted JT k_q	64.84	64.03	64.03	64.03	63.87	63.23	65	64.19	63.71
Embedding $\widehat{\mathbf{e}}$									
Linear	59.19	60.48	58.87	60.48	60.16	60	55.65	55.65	56.13
RBF	64.03	63.87	63.06	64.03	64.52	62.74	63.23	63.55	63.06
Jensen-Shannon	63.39	**64.84**	63.71	60.97	62.74	62.26	66.61	66.13	64.35
Jensen-Tsallis	64.68	**64.84**	64.03	62.74	62.74	63.87	**68.06**	67.1	65.48
Weighted JT $\widetilde{k}_q$	**64.84**	64.03	63.39	64.35	63.55	63.39	65.48	65	63.87
Weighted JT k_q	64.52	64.35	63.87	64.35	**65.65**	63.06	64.68	64.68	63.23

Table 3: Mean accuracy for the best values of q and C for the SVM classifier learnt on ROI 10 using one Rician mixture per class with $K = 4, 5, 6$ components and embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$.

	ROI 10		
No. of components	4	5	6
Embedding $\widetilde{\mathbf{e}}$			
Linear	58.39	58.23	57.42
RBF	66.13	67.26	67.42
Jensen-Shannon	69.68	68.71	68.06
Jensen-Tsallis	**71.13**	70.32	68.87
Weighted JT $\widetilde{k}_q$	70.65	70.97	69.19
Weighted JT k_q	**71.13**	70.32	68.87
Embedding $\overline{\mathbf{e}}$			
Linear	56.29	56.13	55.81
RBF	65.65	67.42	67.26
Jensen-Shannon	68.06	68.55	69.68
Jensen-Tsallis	69.03	69.68	**70.48**
Weighted JT $\widetilde{k}_q$	67.1	67.58	68.39
Weighted JT k_q	67.26	67.26	69.19
Embedding $\widehat{\mathbf{e}}$			
Linear	56.94	57.1	57.9
RBF	67.9	66.94	67.42
Jensen-Shannon	68.55	68.39	69.52
Jensen-Tsallis	69.84	70	**70.48**
Weighted JT $\widetilde{k}_q$	66.94	67.26	68.55
Weighted JT k_q	67.9	67.26	69.03

Table 4: Mean accuracy for the best values of q and C for the set of SVM classifiers obtained by the boosting algorithm, using one Rician mixture per class with $K = 4, 5, 6$ components and embeddings $\widetilde{\mathbf{e}}$, $\overline{\mathbf{e}}$ and $\widehat{\mathbf{e}}$. Results with state-of-the-art methods for ROIs intensity histograms using leave-one-out are also reported.

Boosting			
No. of components	4	5	6
Embedding $\widetilde{\mathbf{e}}$			
Jensen-Shannon	78.55	78.23	77.74
Jensen-Tsallis	79.68	**80.16**	79.03
Weighted JT $\widetilde{k}_q$	80	79.03	78.39
Weighted JT k_q	79.68	**80.16**	79.03
Embedding $\overline{\mathbf{e}}$			
Jensen-Shannon	75	75.97	77.42
Jensen-Tsallis	**78.71**	78.06	79.84
Weighted JT $\widetilde{k}_q$	78.23	78.06	77.58
Weighted JT k_q	**78.71**	78.39	78.55
Embedding $\widehat{\mathbf{e}}$			
Jensen-Shannon	77.90	76.94	76.61
Jensen-Tsallis	79.35	78.39	78.39
Weighted JT $\widetilde{k}_q$	**81.77**	78.39	78.06
Weighted JT k_q	80.48	77.90	78.39

State-of-the-art methods	
Methodology	Accuracy
SVM Best Single ROI	
(Cheng et al., 2009a)	73.4
Dissimilarity representations	
(Ulas et al., 2011)	78.07
SVM Multiple ROIs	
Constellation probab. model + Fisher kernel	
(Cheng et al., 2009b)	80.65
Combined dissimilarity representations	
(Ulas et al., 2010)	79
Dissimilarity representations	
(Ulas et al., 2011)	76.32

a boosting algorithm. The experimental results show that the proposed methodology outperforms the previous state-of-the-art methods on the same dataset.

REFERENCES

Abramowitz, M. and Stegun, I. (1972). *Handbook of Mathematical Functions*. Dover, New York.

Bosch, A., Zisserman, A., and Munoz, X. (2006). Scene classification via plsa. In *Proc. of ECCV*.

Burbea, J. and Rao, C. (1982). On the convexity of some divergence measures based on entropy functions. *IEEE Trans. on Information Theory*, 28(3):489–495.

Cheng, D., Bicego, M., Castellani, U., Cerutti, S., Bellani, M., Rambaldelli, G., Atzori, M., Brambilla, P., and Murino, V. (2009a). Schizophrenia classification using regions of interest in brain MRI. In *IDAMAP Workshop*.

Cheng, D., Bicego, M., Castellani, U., Cristani, M., Cerruti, S., Bellani, M., Rambaldelli, G., Aztori, M., Brambilla, P., and Murino, V. (2009b). A hybrid generative/discriminative method for classification of regions of interest in schizophrenia brain MRI. In *MICCAI09 Workshop on Probabilistic Models for Medical Image Analysis*.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Jour. the Royal Statistical Soc. (B)*, 39:1–38.

Figueiredo, M. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:381–396.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Jour. Comp. System Sciences*, 55:119–139.

Gudbjartsson, H. and Patz, S. (1994). The rician distribution of noisy MRI data. *Magnetic Resonance in Medicine*, 34:910–914.

Jaakkola, T. and Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In *Neural Information Processing Systems – NIPS*.

Lasserre, J., Bishop, C., and Minka, T. (2006). Principled hybrids of generative and discriminative models. In *Proc. Conf. Computer Vision and Patt. Rec. – CVPR*.

Lin, J. (1991). Divergence measures based on Shannon entropy. *IEEE Trans. Information Theory*, 37:145–151.

Martins, A. F., Smith, N. A., Aguiar, P. M., and Figueiredo, M. A. T. (2009). Nonextensive information theoretic kernels on measures. *Journal of Machine Learning Research*, 10:935 – 975.

Ng, A. and Jordan, M. (2002). On discriminative vs generative classifiers: A comparison of logistic regression and naive Bayes. In *Neural Information Processing Systems – NIPS*.

Perina, A., Cristani, M., Castellani, U., Murino, V., and Jojic, N. (2009). A hybrid generative/discriminative classification framework based on free-energy terms. In *Proc. Int. Conf. Computer Vision – ICCV*, Kyoto.

Rice, S. O. (1944). Mathematical analysis of random noise. *Bell Systems Tech. J.*, 23:282–332.

Rubinstein, Y. and Hastie, T. (1997). Discriminative vs informative learning. In *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Suyari, H. (2004). Generalization of Shannon-Khinchin axioms to nonextensive systems and the uniqueness theorem for the nonextensive entropy. *IEEE Trans. on Information Theory*, 50(8):1783–1787.

Ulas, A., Duin, R., Castellani, U., Loog, M., Bicego, M., Murino, V., Bellani, M., Cerruti, S., Tansella, M., and P.Brambilla (2010). Dissimilarity-based detection of schizophrenia. In *ICPR 2010 workshop on Pattern Recognition Challenges in fMRI Neuroimaging*.

Ulas, A., Duin, R., Castellani, U., Loog, M., Mirtuono, P., Bicego, M., Murino, V., Bellani, M., Cerruti, S., Tansella, M., and P.Brambilla (2011). Dissimilarity-based detection of schizophrenia. *Int. Journal of Imaging Systems and Technology*.

APPENDIX

Proof of Proposition 2.1

Proof. First of all, let us note that $f(y_j;\theta_i)$ can be written in factorized form as

$$f_i(y_j;\theta_i) = A(y_j;\theta_i)\cdot B(y_j;\theta_i) \tag{23}$$

where

$$A(y_j;\theta_i) = \frac{y_j}{\sigma_i^2} e^{-\frac{y_j^2+v_i^2}{2\sigma_i^2}} \tag{24}$$

and

$$B(y_j;\theta_i) = I_0\left(\frac{y_j v_i}{\sigma_i^2}\right) \tag{25}$$

It follows that the partial derivatives of the log-likelihhod with respect to v_i and σ_i^2 result

$$\begin{aligned}\frac{\partial \log f(y_j;\theta_i)}{\partial v_i} &= \frac{1}{f(y_j;\theta_i)}\cdot\frac{\partial f(y_j;\theta_i)}{\partial v_i}\\ &= \frac{1}{A\cdot B}\cdot\left[\frac{\partial A}{\partial v_i}\cdot B + A\cdot\frac{\partial B}{\partial v_i}\right]\\ &= \frac{1}{A}\cdot\frac{\partial A}{\partial v_i} + \frac{1}{B}\cdot\frac{\partial B}{\partial v_i}\end{aligned} \tag{26}$$

$$\frac{\partial \log f(y_j;\theta_i)}{\sigma_i^2} = \frac{1}{A}\cdot\frac{\partial A}{\partial \sigma_i^2} + \frac{1}{B}\cdot\frac{\partial B}{\partial \sigma_i^2} \tag{27}$$

The partial derivative of $A(y_j;\theta_i)$ with respect to v_i is

$$\frac{\partial A(y_j;\theta_i)}{\partial v_i} = \frac{y_j}{\sigma_i^2} e^{-\frac{y_j^2+v_i^2}{2\sigma_i^2}}\cdot\left(-\frac{1}{2\sigma_i^2}\cdot 2v_i\right) \tag{28}$$

Moreover, recalling that the higher order modified Bessel functions $I_n(z)$, defined by the contour integral

$$I_n(z) = \frac{1}{2\pi i}\oint e^{\left(\frac{z}{2}\right)\left(\frac{t+1}{t}\right)} t^{-n-1} dt \tag{29}$$

where the contour encloses the origin and is traversed in a counterclockwise direction, can be expressed in terms of $I_0(z)$ through the following derivative identity (Abramowitz and Stegun, 1972)

$$I_n(z) = T_n\left(\frac{d}{dz}\right) I_0(z) \tag{30}$$

where $T_n(z)$ is a Chebyshev polynomial of the first kind (Abramowitz and Stegun, 1972)

$$T_n(z) = \frac{1}{4\pi i}\oint \frac{(1-t^2)t^{-n-1}}{(1-2tz+t^2)} dt \tag{31}$$

with the contour enclosing the origin and traversed in a counterclockwise direction, and in particular that $T_1(z) = z$, then the partial derivative of B results

$$\frac{\partial B(y_j;\theta_i)}{\partial v_i} = \frac{\partial I_0\left(\frac{y_j v_i}{\sigma_i^2}\right)}{\partial v_i} = I_1\left(\frac{y_j v_i}{\sigma_i^2}\right)\cdot\frac{y_j}{\sigma_i^2} \tag{32}$$

Substituting (28) and (32) in (26) we get

$$\frac{\partial \log f(y_j;\theta_i)}{\partial v_i} = -\frac{v_i}{\sigma_i^2} + \frac{I_1\left(\frac{y_j v_i}{\sigma_i^2}\right)}{I_0\left(\frac{y_j v_i}{\sigma_i^2}\right)}\cdot\frac{y_j}{\sigma_i^2} \tag{33}$$

which, substituted in (10) yields (11).

The same considerations hold for the partial derivatives with respect to σ_i^2, yielding to the following expressions for the partial derivative of A and B (with respect to σ_i^2)

$$\frac{\partial A(y_j;\theta_i)}{\partial \sigma_i^2} = -\frac{y_j}{\sigma_i^4} e^{-\frac{y_j^2+v_i^2}{2\sigma_i^2}} + \frac{y_j}{\sigma_i^2} e^{-\frac{y_j^2+v_i^2}{2\sigma_i^2}}\frac{y_j^2+v_i^2}{2\sigma_i^4} \tag{34}$$

$$\frac{\partial B(y_j;\theta_i)}{\partial \sigma_i^2} = I_1\left(\frac{y_j v_i}{\sigma_i^2}\right)\cdot\frac{y_j v_i}{\sigma_i^4} \tag{35}$$

Substituting (34) and (35) in (27), the partial derivative of $\log f(y_j;\theta_i)$ with respect to σ_i^2 results

$$\begin{aligned}\frac{\partial \log f(y_j;\theta_i)}{\partial \sigma_i^2} = &-\frac{1}{\sigma_i^2}\left(1-\frac{y_j^2+v_i^2}{2\sigma_i^2}\right)\\ &-\frac{I_1\left(\frac{y_j v_i}{\sigma_i^2}\right)}{I_0\left(\frac{y_j v_i}{\sigma_i^2}\right)}\cdot\frac{y_j v_i}{\sigma_i^4}\end{aligned} \tag{36}$$

which, plugged in (10) yields (12). □